Systems Modelling Lifecycle (EE5525)

Prof. Peter R Hobson

System Modelling Lifecycle

- Concerned with the question *“How can good systems be built?”*
- For all but the very simplest systems the development process must be managed
  - Planning
  - Monitoring
  - Documentation
  - Toolkits

Terms

- Development process
  - A set of *rules* defining how the process should be carried out, includes how and when documentation should be produced
- Methodology (or method)
  - Techniques for developing artefacts such as a *class model*.
  - Specifies the *modelling language*

Models and modelling languages

Revision!

- *Model*
  - Abstract representation of a *specification, design or system* from a *particular viewpoint*.
- *Language*
  - A way of *expressing* the models produced in the development process
  - Defines a set of *model elements*; models are made up from these (compare a sentence composed of words)
  - It has both *syntax* (rules) and *semantics* (meaning).

Development Process

- Overall High Level process
  - Build and Fix
  - Waterfall (or “throw it over the wall”)
  - Rapid Prototyping
  - Spiral (due to Boehm)
  - OO models

Build and Fix

- No specifications or attempt at design
- Rework until client satisfied
- Maximises the costs of changing the product
- Extreme difficulty in maintenance
- **Don’t use it!**

Waterfall

- A small number of identifiable bits (phases) of the lifecycle
- Activities are sequential in time
- Once a phase is completed it is *never revisited* – in theory
- In practice this is impossible so some level of iteration is required
- Enforced discipline is a positive aspect, also that testing is implicit at every stage of the process
- **Over reliance on documentation**

Analysis and Design

- Analysis
  - Emphasises investigation of the problem and the requirements.
  - Does *not* emphasis a solution
    - Requirements analysis (Use Cases)
    - Object analysis (Domain objects)
- Design
  - Emphasises a conceptual solution *not* its implementation

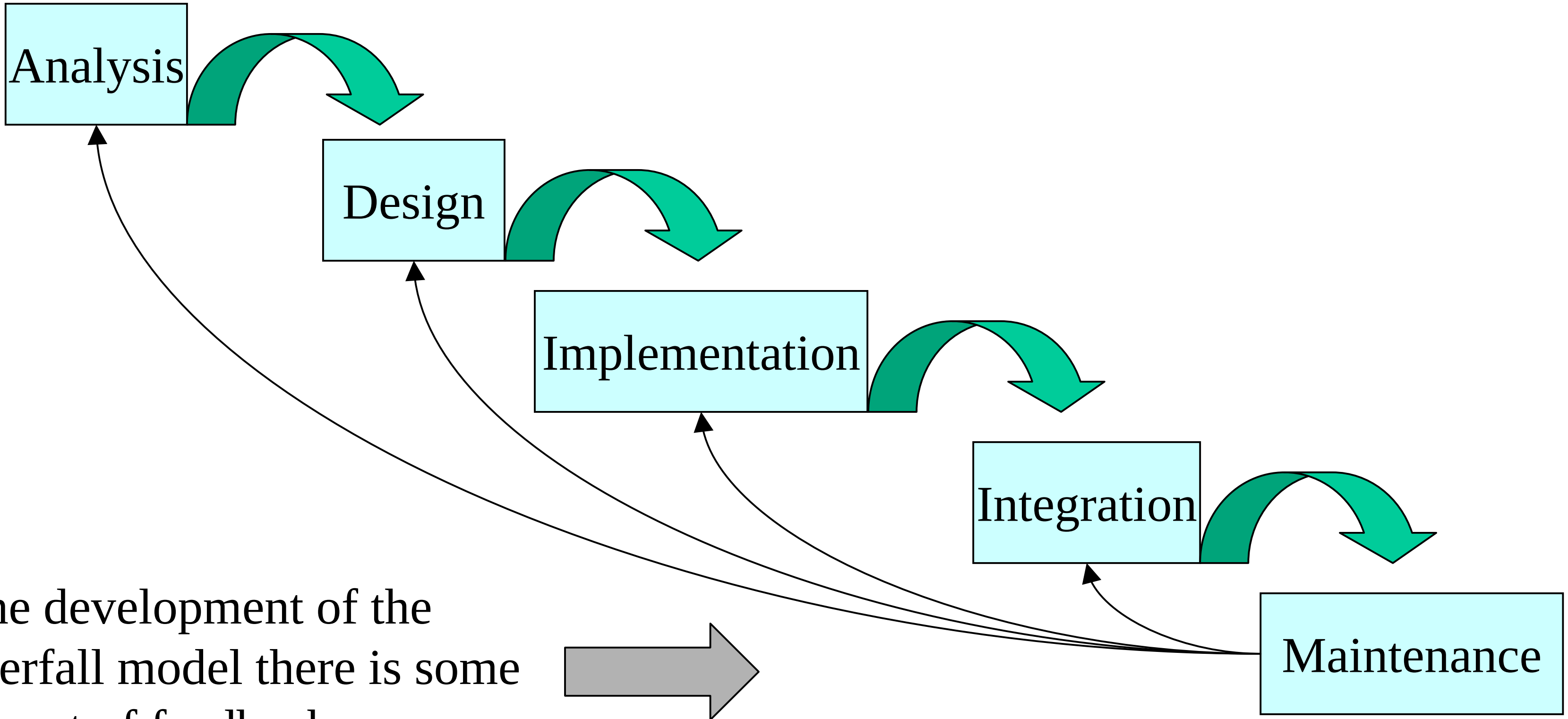
A Catch Phrase!

- Do the right thing (analysis)
- Do the thing right (design)

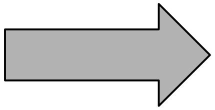
Analysis and Design, but not Implementation, is what **this** part of the EE2161 Course is concerned with.

This makes the concepts and techniques *very general*.

Classical Waterfall



In the development of the Waterfall model there is some Element of feedback



Rapid prototyping

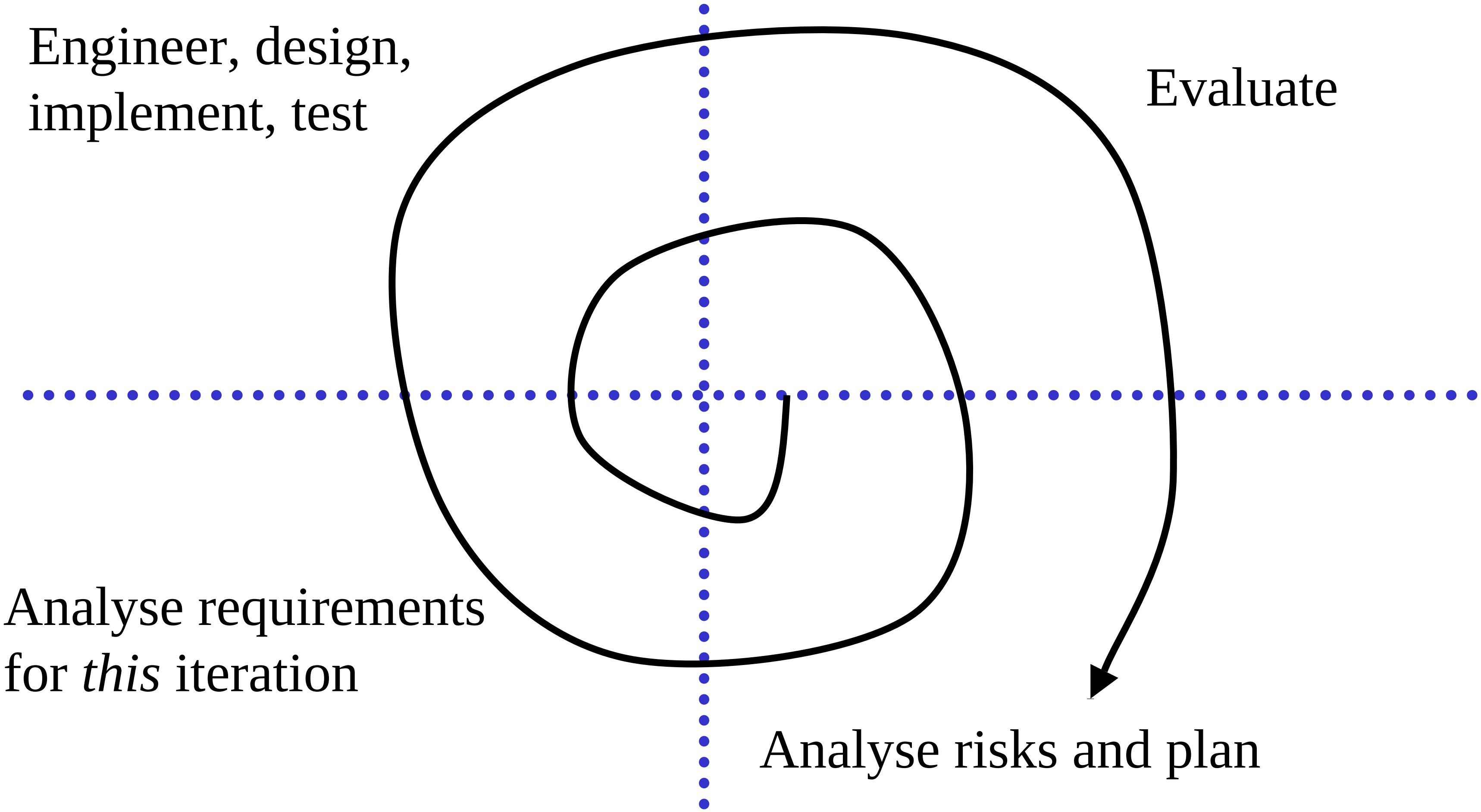
- A working model which is *functionally equivalent* to a subset of the product
- Users evaluate the prototype
- Developers use the feedback from the users to complete the requirements analysis
- Similar to Waterfall model, but with the hope that less feed*back* is needed

Spiral

Make the development process *risk driven*

- Making decisions is essential but risky. The later an error is discovered the more expensive it is to correct. Try to control risk by ensuring errors are discovered early.
- Major risk is misunderstanding the *requirements* – prototyping can help here.
- Not everything can be prototyped (e.g. effect of losing key personnel).

Simple spiral



Engineer, design, implement, test

Evaluate

Analyse requirements for *this* iteration

Analyse risks and plan

Spiral models

- Number of iterations is not fixed
- Risk analysis and prototyping occur several times during the development cycle
- Strengths
  - Emphasis on alternatives and constraints supports reuse
  - Maintenance is built in, equivalent to development
- Weakness
  - Needs expert risk analysis
  - Aimed at internal development of large scale systems

OO

- Iteration
- Parallelism
- Incremental development
- Variety of Models
- Unified Modelling Process

Why iteration?

- Early mitigation of high risk activities
- Visibility of progress
- Rapid feedback from users
- Eases the management of complexity (no “analysis paralysis”)
- Allows the developers to learn and improve the development process

Jargon: OOA, OOD, OOP

- Object Oriented Analysis (OOA)
  - Find and describe the objects or concepts that exists in the *problem* domain.
- Object Oriented Design (OOD)
  - In this phase there is an emphasis on the solution domain objects and their properties and collaboration
  - A design can be implemented
- Object Oriented Programing (OOP)
  - Design objects are implemented

Summary

- Systems modelling lifecycle
  - How do we make good systems?
  - How do we reduce the costs?
  - How do we reduce the chance of failure?
- Approaches
  - Build and Fix
  - Waterfall
  - Spiral
  - *OO*

Resources: http://www.cetus-links.org/oo_ooa_ood_methods.html
WARNING: a good resource but getting old and not updated